UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/782,294	02/14/2001	Andrey Vladimirovich Golchikov	1063-US	2437

24505	7590	06/17/2004

DANIEL J SWIRSKY
PO BOX 2345
BEIT SHEMESH,   99544
ISRAEL

EXAMINER
SON, LINH L D

ART UNIT	PAPER NUMBER
2135	4

DATE MAILED: 06/17/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

	Application No.	Applicant(s)
Office Action Summary	09/782,294	GOLCHIKOV, ANDREY VLADIMIROVICH
	Examiner	Art Unit
	Linh LD Son	2135

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
  Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on <u>14 February 2001</u>.

2a)☐ This action is **FINAL**.   2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) <u>1-18</u> is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) <u>1-18</u> is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All   b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☒ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.

4)☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.

5)☐ Notice of Informal Patent Application (PTO-152)

6)☐ Other: _____.

DETAILED ACTION

Claim Rejections - 35 USC § 103

1.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set
forth in section 102 of this title, if the differences between the subject matter sought to be patented and
the prior art are such that the subject matter as a whole would have been obvious at the time the
invention was made to a person having ordinary skill in the art to which said subject matter pertains.
Patentability shall not be negatived by the manner in which the invention was made.

2.   Claims 1-4, and 7-11 are rejected under 35 U.S.C. 103(a) as being unpatentable

over Cooper et al (US2003/0088515).

3.   As per claim 1, Cooper et al discloses the "Installing and Controlling Trial

Software" invention, which includes method to protecting an executable files

through either of compression and encryption (Para 0032); incorporating a

protection descriptor (Control program, Executable stub, a filter, a key file) into

said executable file, said protection descriptor including information required for

unprotecting said executable file (Para 0032, 0034-0039); providing said

protected executable file to execution apparatus operative to unprotect said

executable file (Para 0027 and 0031); unprotecting said protected executable file

at said execution apparatus using said protection descriptor (Para 0051-0056);

and executing said unprotected executable file at said execution apparatus

(Para 0057-0062). According to Cooper et al invention, the method is to allow

access to the software program files partially until the product is purchased,

instead of protecting the executable files totally. Nevertheless, it is obvious at the time of the invention for one of ordinary skill in the art that Cooper et al's teaching can also implement to protect and prevent access to the executable program files by implementing the methods above until purchased by expiring the trial period (Para 0037).

4.   As per claim 2, Cooper et al discloses a method according to claim 1 wherein said incorporating step comprises including either of a compression key and an encryption key required to uncompress or decrypt said protected executable file in said protection descriptor (Para 0060 and 0063).

5.   As per claim 3, Cooper et al discloses a method according to claim 1 and further comprising encrypting said protection descriptor (Para 0060).

6.   As per claim 4, Cooper et al discloses a method according to claim 1 wherein said providing step comprises providing said protected executable file to an interpreter. By definition of the interpreter in the claimed specification (Page 7), it is used to execute the executable files. Similarly, Cooper et al's invention includes the executable stub (Para 0038) have similar functions to protect the executable file and other programs.

7.   As per claim 7, Cooper et al discloses a method according to claim 1 wherein said providing step comprises providing said protected executable file to a kernel module. On page 7 of the application, the third paragraph describes the steps claimed. Cooper et al does include the same steps. In Para 0065 describes in detail the file composition. In Para 0066-0068, describes the step providing the executable files to the kernel module.

8.   As per claim 8, Cooper et al discloses a method of protecting and executing executable files (See Claim 1 obviousness rejection), the method comprising: protecting at least one function within an executable file through either of compression and encryption (Para 0038 Line 10), thereby creating a protected portion corresponding to said at least one function (Para 0065); preceding said protected portion with a function call instruction to a dynamic unprotector (On the Applicant specification Page 7 cites the dynamic unprotector may be incorporated in the interpreter, which means that the interpreter is also the dynamic unprotector) (In Para 0038 lines 9-20 teaches the interpreter "Executable Stub" and on Para 0042 teaches the interpreter also function as the function call instruction to initiate the unprotection of the executable); executing said function call instruction, thereby executing said dynamic unprotector (Para 0042 and 0062); unprotecting, at said dynamic unprotector (Para 0038, Interpreter, "Executable Stub"), said protected portion, thereby creating an unprotected portion; overwriting said function call instruction and said protected

portion with said unprotected portion; and executing said unprotected portion (Para 0067).

9.   As per claim 9, Cooper et al discloses a method according to claim 8 and further comprising incorporating into said executable file a list identifying said protected function, said list describing any of the function length of said function, the compression method used to protect said function (Para 0065), the encryption method used to protect said function, and a key required to unprotect said protected portion (Para 0064 on page 6), wherein said unprotecting step comprises unprotecting using any information in said list (Para 0065).

10.   As per claim 10, Cooper et al discloses a method according to claim 8 and further comprising providing said executable file to unprotection and execution apparatus, and wherein said executing, unprotecting, and overwriting steps are performed by said unprotection and execution apparatus (Para 0065-0066).

11.   As per claim 11, Cooper et al discloses a method according to claim 10 wherein said protecting step comprises protecting said at least one function within an executable file, and wherein said providing step comprises providing said executable file to an interpreter (See Claim 4 Rejection).

12.   Claims 5-6 and 12 are rejected under 35 U.S.C. 103(a) as being unpatentable

over Cooper et al (US2003/0088515) in view of Li (US/6,334,213).

13.   As per claim 5 and 11, Cooper et al discloses a method according to claim 4.

However, Cooper et al does not teach the executable file is an ELF executable

file and wherein said interpreter is an ELF interpreter.   Nevertheless, Li does

teach the compatibility of the method with Unix platform directly to Common

Object File Format (Col 1 lines 30-48). Therefore, it is obvious at the time of the

invention for one of ordinary skill in the art that Li's Method can also be used to

protect the ELF executable file and the code injection executable is also an ELF

interpreter (Col 5 lines 15-42). Further, the coding is in C++ platform, which is

also operational in Unix (Col 5 line 61)

14.   As per claim 6, Cooper et al discloses a method according to claim 4. However,

Cooper et al does not teach the unprotect step further comprises checking said

protected executable file for the presence of non-standard program code and

unprotect said protected executable file only when said non-standard program

code is present in said protected executable file. Nevertheless, Li does teach

the step to identify the encrypted item to make sure the function item to be

decrypted with no ill-effects (Col 6 lines 46-64). Therefore, it is obvious at the

time of the invention was made for one of ordinary skill in the art that the

identifying step in Li's invention has the capability to check the present of the

non-standard code (encrypted code) in the executable file.

15.   Claims 13-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Li

in view of Ziese (US/6,567,917).

16.   As per claim 13, Li discloses the "Merging of Separate Executable Computer

Programs to Form a Single Executable Computer Program" invention, which

teaches a method of injecting code into an executable file to separate the all the

functions in the executable into items, store the items into a table, encrypt all the

item based on its location (Col 5 lines 15-43, and Col 6 lines 28-46) and store

the necessary decrypting information in the encrypted executable file for later

use. The method is to prevent the executable code from being hacked or

cracked. However, Li does not teach a method of encrypting/decrypting the

function item in the executable by using the cryptographic digest of the function

of the executable code. Nevertheless, Ziese discloses the "Method and System

for Providing Tamper-resistant Executable Software" invention, which teaches a

method of providing tamper-resistant executable files (Col 1 lines 54-59).

Ziese's invention implements the Message digest of the executable file to

encrypt the executable file. The Message Digest then get inserted into the

encrypted executable file and then send to the receiving destination to use the

message digest to decrypt and at the same time provides the integrity

determination of the executable before execute. Therefore, it is obvious at the time of the invention was made for one of ordinary skill in the art to implement Li's executable file injection method to protect a function in the executable by using Ziese's encryption method to control access of the executable and at the same providing integrity checking mechanism.

17.   As per claim 14, Li and Ziese disclose a method according to claim 13 wherein said encrypting step comprises encrypting the address of an instruction that represents the entry point for execution of said executable file (Li, Col 6 lines 10-45).

18.   As per claim 15, Li and Ziese disclose a method according to claim 13 wherein said first hashing, encrypting, and storing steps are performed on a first computer, and wherein said second hashing, decrypting, and executing steps are performed on a second computer (Ziese, Col 4 lines 16-30).

19.   As per claim 16, Li and Ziese disclose a method according to claim 13 and further comprising providing said executable file to unprotection and execution apparatus, and wherein said first hashing, encrypting, and storing steps are performed by said unprotection and execution apparatus (Ziese, Col 3 lines 12-26).

20.   As per claim 17, Li and Ziese disclose a method according to claim 16 wherein

said first hashing, encrypting, and storing steps are performed on an executable

file, and wherein said providing step comprises providing said executable file to

an interpreter (Li, Col 5 lines 15-43).

21.   As per claim 18, Li and Ziese disclose a method according to claim 17.

However, Li and Ziese do not disclose the said executable file is an ELF

executable file and wherein said interpreter is an ELF interpreter. Nevertheless,

Li does teach the compatibility of the method with Unix platform directly to

Common Object File Format (Col 1 lines 30-48). Therefore, it is obvious at the

time of the invention for one of ordinary skill in the art that Li's Method can also

be used to protect the ELF executable file and the code injection executable is

also an ELF interpreter (Col 5 lines 15-42).

Conclusion

22.   Any inquiry concerning this communication from the examiner should be

directed to Linh Son whose telephone number is (703)-305-8914 or Fax to 703-

746-9821.

23.   If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor Kim Y. Vu can be reached at (703)-305-4393. The fax numbers for

this group are (703)-872-9306 (official fax). Any inquiry of general nature or

relating to the status of this application or proceeding should be directed to the

group receptionist whose telephone number is (703)-305-9600.

Linh LD Son

Patent Examiner